



# Touchez Reconnaissance de formes pour l'authentification

13.05.2023

---

Diarra Yacouba, Kanté Younouss Koly, Haïdara Mohamed,

Diabaté Maïmouna, Maïga Maïmouna Henné

@Institut Privée Africain de Technologie et de Management (ITMA)

Baco Djicoroni ACI, Bamako, Mali

Baco Djicoroni ACI, Bamako, Mali

## Aperçu

Ce rapport vise à être un bon point de contrôle sur la progression du projet de recherche sur la reconnaissance des formes tactiles.

Le projet Touch Pattern Recognition est un projet qui aspire à explorer de nouvelles façons d'authentification des systèmes. Ce projet est issu de cinq étudiants de l'université @ITMA [Diarra Yacouba](#), [Kanté Younouss Koly](#), [Haïdara Mohamed](#), [Diabaté Maïmouna](#) et [Maïga Maïmouna Henna](#) dans le cadre de la deuxième édition de « TRANSFORMATION NUMERIQUE », le concours organisé par [Africa\\_Digital](#) en mai 2023.

Ces étudiants ont participé à la ligue universitaire « TRANSFORMATION NUMERIQUE » sur le thème de la sécurité des données.

A noter que ce fichier peut être édité à tout moment en raison de l'état d'avancement du projet et que tous les événements mentionnés dans ce document sont localisés à Bamako/Mali entre le 30 avril et le 13 juillet 2023.

Cela dit, l'idée derrière le projet est de remplacer les informations d'authentification traditionnelles du système (nom d'utilisateur/mot de passe) par un modèle d'apprentissage profond (DL) formé pour reconnaître l'utilisateur du système via des données tactiles sur des appareils tactiles.

À cette fin, les données sont représentées par rapport à différents champs tels que les positions, les vitesses, les directions, la latence (délai) de chaque contact.

Ce document va parler du travail effectué par les étudiants cités précédemment, de l'acquisition des données au déploiement du modèle en passant par les problématiques rencontrées.

## Objectifs

Ce projet vise à :

1. Repenser notre façon de concevoir les systèmes d'authentification.
2. Apportez plus d'innovation et de sécurité aux systèmes d'authentification.
3. Facilitez l'accès aux systèmes en remplaçant les mots de passe qui peuvent être fastidieux car ils sont faciles à oublier ou à pirater.
4. Recherchez de nouvelles applications des systèmes d'IA qui deviennent de plus en plus nombreuses puissant.

## Acquisition et prétraitement des données

Nous allons parler de la manière dont les données ont été collectées et traitées pour la formation.

### I. Acquisition de données

Tout d'abord, nous avons utilisé une application Web mobile existante appelée « TouchTracker » développée par Marco Cancellieri. Cette application dispose d'une interface utilisateur simple qui affiche un écran vert et enregistre tous les gestes effectués sur l'écran écrits en JavaScript.

Plus d'informations sur : [TouchTracker](#)

Cependant, le but initial de l'application n'était pas de collecter les données tactiles dont nous avons besoin, nous avons donc apporté des modifications à son code source pour collecter les données nécessaires. En particulier, nous avons modifié le code de l'application pour collecter les positions tactiles, les vitesses, les directions, les durées et les horodatages.

Ces modifications nous ont permis de recueillir des informations détaillées sur les gestes tactiles effectués sur l'écran, ce qui est essentiel pour entraîner un modèle d'apprentissage automatique/Deep Learning à reconnaître et à faire des prédictions avec précision.

Une fois les données collectées, elles seront exportées dans des fichiers .json qui est un format de données structuré facilement lisible et traité par la plupart des langages de programmation, dont Python, le langage que nous utilisons pour entraîner vos réseaux neuronaux.

Pour poursuivre le processus d'acquisition de données, nous avons créé une interface utilisateur pour notre application Web basée sur TouchTracker, en ajoutant du code HTML, CSS et JS.

HTML, CSS et JS sont les trois technologies fondamentales utilisées pour créer des sites Web et des applications Web. HTML fournit la structure de la page, CSS contrôle la présentation et la mise en page, et JS gère l'interactivité et les fonctionnalités.

\*Remarque : Ce rapport peut être lu par des personnes n'ayant aucune expérience des domaines concernés par notre travail, c'est pourquoi nous introduisons des technologies comme HTML qui sont bien connues de C'est les gars\*

Nous devons modifier le code JS pour gérer les événements tactiles. Cela impliquait probablement de réécrire ou de modifier les auditeurs d'événements pour reconnaître les gestes tactiles tels que les tapotements, les glissements et les pincements, et pour collecter les données tactiles dont nous avons besoin pour la formation.

Dans l'ensemble, notre processus d'acquisition de données impliquait la modification des applications Web mobiles existantes, la collecte de données tactiles détaillées et l'exportation des données dans un format compatible.

Notez qu'à la date de rédaction de ce rapport, la demande n'était pas complètement terminée en raison de problèmes qui seront décrits dans la dernière section de ce document.

## II. Prétraitement

Après l'acquisition des données, nous avons dû représenter les données sous une forme appropriée pour former un réseau neuronal artificiel (vous pouvez penser à ceux-ci comme une abstraction des réseaux neuronaux du cerveau humain pour le calcul de haut niveau). Pour cela nous avons écrit un script python qui lit chaque fichier .json téléchargé et extrait les champs qui nous intéressent tels que les positions (liste de 2 listes d'éléments dont la première est Position.X et la seconde Position.Y par rapport à un objet fictif point de référence sur l'écran), les vitesses (liste des vitesses euclidiennes calculées pour chaque nouvelle position enregistrée par rapport au point de départ du toucher), les directions (liste des directions/angles de chaque toucher par rapport au précédent en radians) et les horodatages que nous utilisons pour calculer la durée de chaque contact.

Notez que les positions ont été divisées en deux nouvelles listes position.X et position.Y.

Et puis, afin d'obtenir les caractéristiques qui seront utilisées pour entraîner le modèle, nous avons calculé la moyenne, la déviation standard, le min, le max et la plage de chaque champ ou liste (Position.X, Position.Y, vitesses, directions) et ajouté la durée du contact.

De cette façon, nous avons obtenu une nouvelle représentation de nos données qui est un [numpy.array](#) de forme (21,) qui est une assez bonne représentation des données pour la formation d'un modèle de Deep Learning.

Les données ont ensuite été étiquetées et enregistrées avec [Python.pickle](#).

NB : L'étiquette était des valeurs binaires (0, 1)

## Formation et évaluation en IA

### I. Création & Formation

Nous avons décidé d'opter pour un modèle Keras.

Keras est une bibliothèque Python pour l'apprentissage en profondeur qui englobe les bibliothèques numériques efficaces [Tensorflow](#) et [Théano](#). Keras vous permet de concevoir et de former rapidement et simplement des réseaux de neurones et des modèles d'apprentissage profond.

Nous créons donc un [Keras Sequential](#) assez simple modèle.

Les poids sont initialisés à l'aide d'un petit [nombre aléatoire gaussien](#). La fonction d'activation du redresseur est utilisée. La couche de sortie contient un seul neurone afin de faire des prédictions. Il utilise la fonction d'activation sigmoïde afin de produire un

sortie de probabilité dans la plage de 0 à 1 qui peut facilement et automatiquement être convertie en valeurs de classe précises.

Enfin, nous avons utilisé la [fonction de perte logarithmique \(entropie croisée\)](#) pendant l'entraînement, l'une des fonctions de perte les plus efficaces pour les problèmes de classification binaire. Le modèle utilise également l'optimiseur efficace [Adam](#) pour [la descente de pente](#), et des mesures de précision seront calculées pendant la formation.

Le modèle n'avait qu'une seule couche cachée de 10 neurones. La topologie du modèle peut donc être résumée par [21, 10, 1]. Il n'y avait que 693 paramètres

## II. Évaluation et sauvegarde

Nous avons d'abord évalué les topologies de modèles de couple en utilisant la validation croisée stratifiée du framework scikit-learn.

L'idée était d'utiliser cette étape comme un test consistant en un entraînement de courte durée et de les évaluer afin de choisir le meilleur modèle pour s'entraîner plus longtemps.

La validation croisée stratifiée ([StratifiedKFold](#)) est une technique de rééchantillonnage qui fournira une estimation des performances du modèle. Pour ce faire, il divise les données en k parties et entraîne le modèle sur toutes les parties sauf une, qui est présentée comme un ensemble de test pour évaluer les performances du modèle.

Nous avons été vraiment surpris par les performances du modèle, même sans aucune couche cachée, qui a obtenu une précision de 83 %. Cela soulève une fois de plus la question de la qualité des données. Ne nous sommes-nous pas entraînés avec des données biaisées ?

Après une formation continue de seulement 1 000 époques, la topologie [21, 10, 1] a atteint une précision de 90 % et c'est le modèle que nous avons enregistré au moment de la rédaction de ce document (13 mai 2023).

Nous avons ensuite enregistré les modèles pré-entraînés prêts pour l'inférence, le test ou la formation complémentaire dans différents formats :

- Dans un dossier

Le format SavedModel est un répertoire contenant un binaire protobuf et un point de contrôle TensorFlow. La première approche utilisée pour sauvegarder le modèle

- Au format TF.js Layers

Nous avons enregistré le modèle au format TF.js Layers après un bref test pour l'utiliser ultérieurement avec l'application

## Déploiement du modèle

Nous avons déjà exporté le modèle vers [Tensorflow.js](#) format de couches et l'a intégré avec succès à l'application web utilisée pour collecter les données. Bien que certains problèmes subsistent, comme la précision du modèle qui diminue une fois déployé ou l'inférence sur les iPhones, nous avons pu tester notre modèle entraîné sur des androïdes sans de très bons résultats.

### Problèmes

Dans cette section, nous allons parler de deux problèmes ou tâches qui n'ont pas encore été corrigés.

#### I. Amélioration de l'interface utilisateur de

**l'application** Nous souhaitons améliorer l'esthétique de l'application et éventuellement ajouter quelques fonctionnalités pour améliorer l'expérience utilisateur lors des tests du modèle, même si l'application Web n'est pas essentielle au projet.

#### II. Corriger les bugs sur les iPhones

Nous devrions bientôt résoudre les problèmes liés aux inférences sur les iPhones ou proposer une application spécialement conçue pour les iPhones.

## Résumé et notes

Pour résumer ce document, nous présentons une petite idée de projet de recherche explorée par cinq étudiants au Mali.

Avec les résultats décrits ici, nous pensons que ce projet a un grand potentiel et aimerions intéresser les gens. Mais nous ne pouvons pas affirmer l'efficacité, le prix abordable et la fiabilité de tels systèmes s'ils étaient réellement complets.

Il n'y a rien de spectaculaire pour l'instant mais nous espérons que ce sera le cas. Même si nous allons probablement arrêter notre travail sur ce projet pour le moment, nous espérons que nous-mêmes et/ou d'autres trouverons de nouvelles idées pour améliorer l'idée de base du projet et le travail effectué.

Vous pouvez obtenir tous les codes écrits à cette date sur ce [github](#) et contactez les auteurs avec les liens suivants :



Kanté Younouss Koly : [email](#)

Haïdara Mohamed : [email](#), [Facebook](#), [LinkedIn](#),

Diabaté Maïmouna : [email](#)

Diarra Yacouba : [email](#), [Facebook](#), [LinkedIn](#), [Twitter](#), [github](#), [Débordement de pile](#)

Maïga Maïmouna Henné : [email](#), [Twitter](#), [LinkedIn](#)